

# Flooding-Assisted Threshold Assignment for Aggregate Monitoring in Sensor Networks

Ali Abbasi<sup>1</sup>, Ahmad Khonsari<sup>1,2</sup>, Mohammad S. Talebi<sup>2</sup>

<sup>1</sup> ECE Department, University of Tehran, Iran

<sup>2</sup> School of Computer Science, IPM, Iran

a.abbasi@ece.ut.ac.ir, {ak,mstalebi}@ipm.ir

**Abstract.** The research community has witnessed a large interest in monitoring large scale distributed systems. In these applications typically we wish to monitor a global system condition which is defined as a function of local network elements parameters. In this paper, we address *Aggregate Threshold Queries* in sensor networks, which are used to detect when an aggregate value of all sensor measurements crosses a predetermined threshold. The major constraint in designing monitoring applications is reducing the amount of communication burden which is the dominant factor of energy drain in wireless sensor networks. In this study, we address the aggregate threshold monitoring problem by proposing a distributed algorithm to set local thresholds on each sensor node so as to minimize the probability of global polling. We adopt the FPTAS optimization formulation of the problem [2] and propose a distributed algorithm as the solution to the problem. Simulation results demonstrate the validity of the proposed distributed algorithm in attaining very close performance as the centralized schema.

## 1 Introduction

In recent years, with the abundance of emerging large scale distributed systems such as sensor networks, and peer to peer networks, monitoring scenarios is progressively more considered vital to track these systems. In monitoring either we are interested in supervising the network itself (e.g. traffic engineering, routing optimization, anomaly detection in data networks), or environment that network deployed in it (e.g. wildlife behavior, moving objects in sensor networks). Network monitoring includes measuring system parameters to react to different network conditions. There are two ways of getting knowledge from the network; the first is to send requests into the network to poll all relevant information. Another option is that all network elements push all possibly important readings to the management station. The nature of network monitoring applications imposes specific requirements on design of monitoring algorithm: 1- the algorithm should provide real time detection of noteworthy events 2- The algorithm must be scalable to large number of nodes. 3- The detection process should incur minimum communication in the network. This is a very critical requirement in sensor networks, because communication is the primary source of energy consumption in these networks. Sensors are powered by batteries and replacing these

if not impossible is usually cumbersome. It is also important in wired networks. In these Networks, the monitoring application should not hamper the normal operation of the network. Thereby, a challenge in monitoring applications is to devise plans to reduce communication while fulfilling application requirements. A common method is to install local constraints or filters at remote sources to filter out unnecessary updates. Local constraints should have the property such that preserving all of them ensures that there was not any anomalous event. Each new measurement is compared to the filter and in case it violates (exits) the corresponding range of the filter an update is sent to the base station. Apparently efficient decomposition of global system constraint into local constraint has a great impact on reducing communication. Filter setting is dependent on a number of parameters including the associated cost with sending an update from each node to the management station and the changing patterns of measurements. Often the monitoring task is the need to detect when a function of individual network element readings crosses a given threshold. Typically, monitored functions are aggregate functions like SUM, or AVG which gives global insights concerning the state of the network. These functions are specially important in sensor networks where individual sensor readings are inherently unreliable and don't convey much information. Consider the following queries:

- Report when the number of enemy troops detected in a region of the network crosses 20.
- Report when variance of temperature in a building crosses 3.

Aggregate functions like SUM, AVG are linear functions of individual sensor readings, thus the global property of  $\sum_i A_i T_i \leq T$  can be decomposed to a set of  $X_i \leq T_i$  constraints which checked locally at each node. In this work, we concentrate on the problem of determining optimal values for  $T_i$ 's. Previous works [2] [18] assume that the monitoring station computes the optimal thresholds based on individual nodes probability distribution functions and assigns thresholds for nodes. Thus each node should update its histogram constructed over recent measurements either periodically or based on a change detection algorithm to the base station. But there is limitation on the applicability of such an approach for sensor networks. The structure of such networks has necessitated the design of asynchronous, distributed and fault-tolerant computation and information exchange algorithms [5].

This is mainly because of sensor networks operational constraints: (i) the network topology may not be completely known to the nodes of the network, (ii) nodes may join or leave the network (even expire), so that the network topology itself may change. These issues specially occurs in mobile sensor networks. In this work, we propose a distributed algorithm for optimal threshold assignment so as to minimize the probability of global system polling.

The remainder of the paper is organized as follows: Section 2 reviews related works. Section 3 defines the system model and problem definition. Section 4 provides the optimal solution for the threshold assignment problem. Section 5 presents a distributed threshold assignment algorithm based on the problem's

optimal solution. Section 6 validates experimental evaluation of the proposed method. Finally section 7 concludes the study.

## 2 Related Works

In recent years, continuous query processing for monitoring distributed data streams has attracted much research. Different query types have been mentioned in this environment includes top-k[3], quantiles[7], joins[6]. Monitoring aggregate threshold queries in networks initially mentioned by the pioneering work of Raz et. al [12]. They introduced installing local mathematical constraints at remote sites and present a simple approach for threshold assignment assuming uniform data distributions of system variables. Olston et. al [18] suggested an adaptive filter based approach for tracking error bounded values of aggregate functions. According to the precision specified in continuous query filters are assigned for local values of data objects. Keralpera et. al[16] presented several algorithm for static and adaptive threshold setting for monitoring thresholded counts queries and analyzed the communication complexity of each algorithm. Sharfman et al[20] introduced a geometric approach for monitoring arbitrary threshold functions. Recent work of kashyap et al[14] considers the problem of non-zero slack threshold assignment which adaptively dedicate fraction of total threshold to monitoring node to absorb small local threshold crossing that eliminate the need to global system polling. All the above works assume centralized threshold computation and assignment but in this paper we propose a distributed algorithm for optimal threshold setting.

## 3 System Model and Problem Formulation

We assume that there is a base station being responsible for monitoring the network. The base station disseminates queries to sensors and responds to user queries. Each sensor node  $i$  continuously reads the requested local phenomenon  $x_i$  which is in the range  $[0, M_i]$  at a fixed sampling rate. Amongst the local phenomenon are temperature, humidity, light, etc. The measured values of all sensors can be represented in a vector of values  $\mathbf{x} = (x_i, i = 1..n)$ .

The goal is to detect when the aggregate value of all the measurements  $\sum_{i=1}^n A_i x_i$  crosses a predetermined threshold  $T$ . A common method to reduce communication is to install local constraints or filters (in the form of an interval  $I_i = [0, T_i]$ ) at remote sources. Local constraints setting should maintain the property that preserving all of them, ensures that the aggregate value has not exceeded the threshold (covering property). Each sensor  $i$  after measuring the new value of  $x_i$ , checks condition  $L_i \equiv (x_i \in [0, T_i])$ , and the condition upon being violated, an update is sent to the base station to initiate global aggregate computation. Using such a method, vast amount of updates are filtered out at the source and are not transmitted to the base station. The efficiency of this approach is largely dependent on filter setting method. But what values should we choose for  $T_i$ 's? Selection of values must satisfy the global property

$\sum_{i=1}^n A_i T_i \leq T$  to conform covering property. However, this equation leads to great flexibility in choosing  $T_i$ 's. In this respect, we face to another question: what is the best selection?

We aim at maximizing the communication cost via minimizing the probability of global polling. Every local filter violation leads to global computation, thus we should maximize the probability of preserving all local constraints  $\max F(x_1 \leq T_1, \dots, x_n \leq T_n)$ , where  $F$  is the joint cumulative frequency distribution over all the sensors. Computing the multi-dimensional histograms is somewhat cumbersome in terms of communication burden, and hence we consider more communication convenient assumption of independence between sensor's data distributions. With this simplifying assumption, we come up to the optimization problem

$$\max_{\mathbf{T}} \prod_{i=1}^n F_i(T_i) \quad (1)$$

subject to:

$$\sum_{i=1}^n A_i T_i \leq T \quad (2)$$

where  $\mathbf{T} = (T_i, i = 1..n)$  is the vector of local thresholds. It has been proven that local threshold selection problem is *NP-hard* [2] wherein the authors have introduced a centralized scheme, referred to as FPTAS. This centralized scheme was used to solve the problem within  $\epsilon$  relative error for arbitrarily small  $\epsilon$ . The solution was based on the assumption that all the values of  $F_i(T_i)$  are integral powers of a constant  $\alpha$ , which is assumed to be slightly greater than 1. Thus, each  $F_i(T_i)$  will correspond to some  $\alpha^{r_i}$  and maximizing  $\prod_{i=1}^n F_i(T_i)$  will be equivalent to maximizing  $\alpha^{\sum r_i}$ . If  $T_i(r)$  denotes the local threshold value such that

$$\alpha^{r_i} \leq F_i(T_i(r_i)) < \alpha^{r_i+1} \quad (3)$$

the problem becomes

$$\max_{\mathbf{r}} \sum_{i=1}^n r_i \quad (4)$$

subject to:

$$\sum_{i=1}^n A_i T_i(r_i) \leq T \quad (5)$$

where  $\mathbf{r} = (r_i, i = 1..n)$  is the vector representation of powers.

This problem is a variant of *knapsack problem* that can be solved using dynamic programming.

In order for (4) to admit a unique maximizer,  $F_i$  must satisfy the following assumptions:

- A1:**  $F_i$  is positive, strictly increasing and twice-continuously differentiable.
- A2:**  $F_i$  is log-concave, i.e.  $\log F_i$  is a concave function.

Indeed, we assume that optimization variable,  $\mathbf{r}$ , belongs to a domain in which  $F_i$  satisfies the abovementioned assumptions.

## 4 Optimal Solution

In this section, we solve problem (4). Problem (4) is a constrained problem, whose constraint (5) is coupled across the network. Such a constrained optimization problem can be efficiently solved using Interior Point Method [5], which necessitates the coordination among possibly all nodes of the networks, which is undesirable or infeasible. However, in the context of wireless ad-hoc and sensor networks, we are interested in distributive algorithms to solve (4).

Towards this end, we aim at solving the problem (4) through its dual. In the sequel, we proceed to derive the dual problem of (4) and then present a distributive iterative algorithm as the solution to the dual problem.

We start by writing the Lagrangian of problem (4), as follows

$$L(\mathbf{r}, \mu) = \sum_{i=1}^n r_i - \mu \left( \sum_{i=1}^n A_i T_i(r_i) - T \right) \quad (6)$$

where  $\mu > 0$  is the Lagrange multiplier associated with constraint (5). Using Karush-Kuhn-Tucker (KKT) conditions for convex optimization, to find optimal powers  $\mathbf{r}^*$ , we should find the stationary points of the Lagrangian and satisfy complementary slackness conditions. The complementary slackness conditions for optimal primal variable  $\mathbf{r}^*$  and dual variable  $\mu^*$ , are

$$\mu^* \geq 0; \quad (7)$$

$$\sum_{i=1}^n A_i T_i(r_i^*) \leq T; \quad (8)$$

$$\mu^* \left( \sum_{i=1}^n A_i T_i(r_i^*) - T \right) = 0 \quad (9)$$

In order to find the stationary points of the Lagrangian, we solve

$$\nabla L(\mathbf{r}^*, \mu^*) = \mathbf{0} \quad (10)$$

where  $\mathbf{0}$  is a vector with all zero. For the  $i$ th element of (10) we have

$$\frac{\partial L}{\partial r_i} = 1 - \mu A_i \frac{dT_i(r_i)}{dr_i} \quad (11)$$

From (3), recall that  $T_i(r_i)$  is selected so that  $\alpha^{r_i} \leq F_i(T_i(r_i)) < \alpha^{r_i+1}$ , also, recall that  $\alpha$  is a constant slightly greater than 1. Therefore, the lower and upper bounds of  $F_i(T_i(r_i))$  are sufficiently close to each other, leading us to approximate  $F_i(T_i(r_i))$ , using the concept of geometric mean as following:

$$\begin{aligned} F_i(T_i(r_i)) &\approx \sqrt{\alpha^{r_i} \alpha^{r_i+1}} \\ &\approx \sqrt{\alpha} \alpha^{r_i} \end{aligned} \quad (12)$$

Such an approximation would also be done through the concept of average mean, which for  $\alpha = 1 + \delta$ ,  $\delta \rightarrow 0$  gives the same results, as in (12).

Recalling the monotonicity assumption of  $F_i$ , it admits a unique inverse, which yields the explicit expression for  $T_i$  as

$$T_i(r_i) = F_i^{-1}(\alpha^{r_i} \sqrt{\alpha}) \quad (13)$$

Substituting (13) in (11), yields

$$\frac{\partial L}{\partial r_i} = 1 - \mu A_i \frac{dF_i^{-1}}{dr_i} (\alpha^{r_i} \sqrt{\alpha}) \quad (14)$$

$$\begin{aligned} &= 1 - \mu A_i \frac{d}{dr_i} (\alpha^{r_i} \sqrt{\alpha}) \left. \frac{dF_i^{-1}}{dr_i} \right|_{\alpha^{r_i} \sqrt{\alpha}} \\ &= 1 - \mu A_i \alpha^{r_i} \sqrt{\alpha} \ln \alpha \left. \frac{dF_i^{-1}}{dr_i} \right|_{\alpha^{r_i} \sqrt{\alpha}} \end{aligned} \quad (15)$$

Setting (15) to zero and doing some algebraic manipulation, gives an explicit expression for the optimal exponent  $\mathbf{r}^*$ , in terms of optimal Lagrange multiplier  $\mu^*$  and other network parameters. For the sake of presentation, we define

$$G_i(z) = \left. \frac{dF_i^{-1}}{dr_i} \right|_z \quad (16)$$

Substituting  $G_i(\cdot)$  in (15), we come up to the following implicit equation to obtain  $\mathbf{r}^*$

$$\alpha^{r_i^*} G_i(\alpha^{r_i^*} \sqrt{\alpha}) = \frac{1}{\mu^* A_i \ln \alpha \sqrt{\alpha}} \quad (17)$$

In order to solve problem (4) through its dual, we need to obtain the Lagrange dual function, or simply dual function. The Lagrange dual function  $D(\mu)$  is defined as the maximum of the Lagrangian  $L(\mathbf{r}, \mu)$  over the primal variable  $\mathbf{r}$ , for a given  $\mu$ . Thus,  $D(\mu)$  can be expressed as

$$D(\mu) = \max_{\mathbf{r}} L(\mathbf{r}, \mu) \quad (18)$$

Based on the results of the KKT condition mentioned above, maximization in (18) is already solved with  $\mathbf{r}^*$  given by (17), which results in

$$D(\mu) = L(\mathbf{r}^*, \mu) \quad (19)$$

The dual problem is formulated as

$$\min_{\mu} D(\mu) \equiv \min_{\mu} L(\mathbf{r}^*, \mu) \quad (20)$$

subject to:

$$\mu \geq 0 \quad (21)$$

Dual problem defined above can be solved using iterative methods. In order to obtain a distributed algorithm, we solve the dual problem (20) using Gradient Projection Method. To solve the dual problem, Gradient Projection Method

adjusts  $\mu$  in opposite direction to the Gradient of dual function, i.e.  $\nabla D(\mu)$ . Precisely speaking, in the  $k$ th iteration step,  $\mu^{(k)}$  is updated as follows

$$\mu^{(k+1)} = \left[ \mu^{(k)} - \gamma \frac{dD(\mu^{(k)})}{d\mu} \right]^+ \quad (22)$$

where  $[z]^+ = \max\{z, 0\}$  and  $\gamma$  is a sufficiently small constant step size. Using the Danskin's Theorem [4], the derivative of  $D(\mu)$  is given by

$$\frac{dD(\mu)}{d\mu} = T - \sum_{i=1}^n A_i T_i(r_i) \quad (23)$$

Substituting (23) in (22), yields

$$\mu^{(k+1)} = \left[ \mu^{(k)} + \gamma \left( \sum_{i=1}^n A_i T_i(r_i^{(k)}) - T \right) \right]^+ \quad (24)$$

where  $r_i^{(k)}$  is the solution to (17) for a given  $\mu^{(k)}$ . In this equation,  $\gamma$  is chosen sufficiently small so as to guarantee the convergence.

In the economics literature, Lagrange multiplier or dual variable,  $\mu$  is called *shadow price* [15] and accordingly, (24) is called shadow price update. This stems from the interpretation of its role in solving the primal problem via its dual. From (17) it's apparent that  $\mathbf{r}^*$  is a decreasing function of  $\mu$ ; therefore  $\mu$  can be construed as the price which must be paid by node  $i$  to achieve the threshold  $T(r_i)$ . As the nature of such a price is hidden to the sources from the primal problem perspective, it is called *shadow price*.

(17) and (24) can be utilized as an iterative solution to problem (20) and thereby (4). At each iteration step  $k$ , dual variable  $\mu$  will be updated based on the history of itself and the primal variables  $\mathbf{r}$ . Then, it would be utilized by to update primal variable  $\mathbf{r}$ , accordingly. Therefore, after spending enough iteration steps, primal and dual variables tends to primal-optimal  $\mathbf{r}^*$  and dual-optimal  $\mu^*$ , respectively.

Based on the above iterative solution, we propose a distributed algorithm as a solution to threshold selection problem. We defer the algorithm until the Section 5.

## 5 Distributed Threshold Assignment Algorithm

In this section, we propose a distributed algorithm based on the iterative solutions obtained in Section 4. Considering (24) and (17), it is clear that the iterative solution to problem (4) can be regarded as a distributed algorithm.

The algorithm is devised by directly utilizing the update equation (17) and (24), over the network. Clearly, for  $\mu$  to be updated using (24), the knowledge about the evolution of all nodes in the network is required. Although the nature of solution is distributive, this is a global information of the network and can be elaborated using well-known algorithms such as Flooding, etc.

In particular, each sensor  $i$  in the iteration step  $k$  benefits from all other nodes' current power  $r_{-i}$ 's, thanks to Flooding-like algorithms, to update the current shadow price  $\mu^{(k)}$ . Since all other nodes have access to such information too, they will obtain the same value for  $\mu^{(k+1)}$  and therefore, we don't introduce additional notation to distinguish between the realized update process.

Upon updating  $\mu^{(k)}$ , each sensor  $i$  calculates its power  $r_i$ , accordingly. The above rule will proceed until reaching some predefined notions of convergence.

We will refer to this algorithm as DTA. The DTA Algorithm is stated below.

---

**DTA** Distributed Threshold Assignment Algorithm

---

**Initialization**

Initialize  $A_i$ s and  $T \forall i = 1..n$ .

**Main Loop**

Do until  $\max_i |r_i^{(k+1)} - r_i^{(k)}| < \epsilon$

1. t each sensor node, update the shadow price as following:

$$\mu^{(k+1)} = \left[ \mu^{(k)} + \gamma \left( \sum_{i=1}^n A_i T_i(r_i^{(k)}) - T \right) \right]^+$$

2. Update  $r_i$  according to the following equation:

$$\alpha^{r_i^{(k)}} G_i(\alpha^{r_i^{(k)}} \sqrt{\alpha}) = \frac{1}{\mu_i^{(k+1)} A_i \ln \alpha \sqrt{\alpha}}$$


---

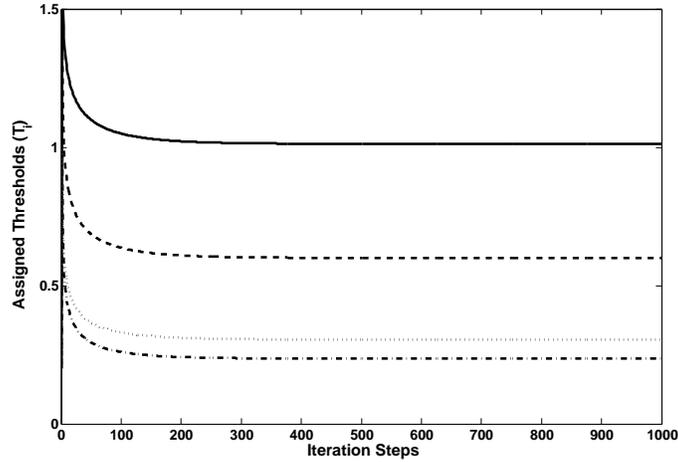
**DTA.** Distributed Threshold Assignment Algorithm

## 6 Experimental Evaluation

We have conducted simulation experiments to evaluate the performance of our proposed algorithm. We verify that the algorithm, locally executed on each node, may indeed achieve the desired global optimal threshold assignment.

In our simulation scenario, we consider a sensor network consisting of 100 sensor nodes which randomly scattered over an area.

We assume that each node  $i$  incessantly takes measurements of a physical phenomenon, whose CDF obeys an exponential distribution with the exponent parameter  $\lambda_i$ . Although such a distribution may sound to be of limited interest, it is worth mentioning that many significant real world applications might fall within such a framework. Amongst such applications are monitoring the dwell time of a traffic flow which pursue a Poisson distribution. The corresponding coefficient  $A_i$  is assumed to be randomly drawn from a uniform distribution over  $[0, 5]$ . Step size is chosen to be  $\gamma = 1.2$  and the total threshold  $T$  is set to 10.



**Fig. 1.** Evolution of The Assigned Thresholds  $T_i(r_i)$  for Some Nodes Using DTA Algorithm

The most significant issues of interest are the evolutions of primal and dual (shadow price) variables. Evolution of assigned thresholds  $T_i$ s and shadow price  $\mu$  for DTA Algorithm are depicted in Fig. 1 and 2, respectively. It is apparent from these figures that by spending less than 300 iteration steps, convergence was achieved and thereafter,  $\mu$  and  $T_i$ s had intangible variations.

## 7 Conclusion

There have been many studies exploring various applications of WSNs such as monitoring. In the context of WSNs, we are interested in the design of asynchronous, distributed algorithms for exchanging the information among the sensor nodes. In this paper, we focused on the problem of distributed threshold selection for aggregate threshold monitoring in WSNs. Towards this end, we formulated threshold selection as an optimization problem that considers the cumulative distribution function of distinct monitoring variables. The fundamental objective of the optimization problem was to minimize the probability of global polling. The original problem was non-convex, thus we adopted the so called FPTAS reformulation which was convex and has been solved using a centralized approach. We elaborated their method and solved the problem via its dual so as to achieve a distributed solution. Our distributed solution leads to a distributed algorithm; called DTA Algorithm which acts based on the global evolution of the network information. Such information is gathered by a flooding-like algorithm. The results extracted from the experimental evaluation was promising and demonstrated the achieved performance of the suggested algorithm are quite comparable to the results of the centralized approach.

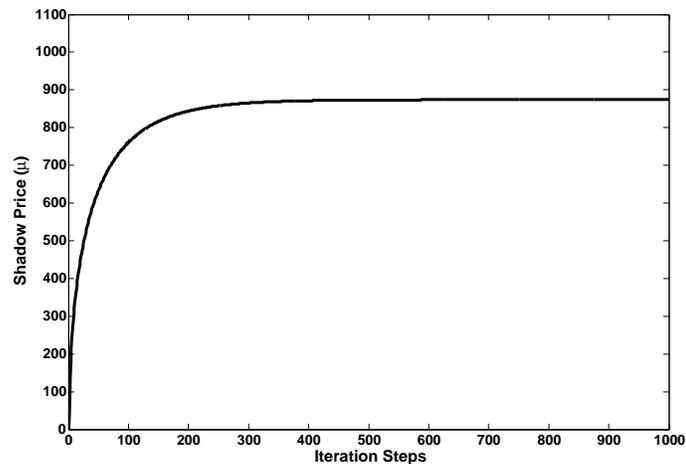


Fig. 2. Evolution of The Shadow Price  $\mu$  Using DTA Algorithm

## References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A Survey on Sensor Networks, *IEEE Communications Magazine*. **40**(8) (2002) 102–114
2. Aggrawal, S., Deb, S., Naidu, K. V. M., Rastogi, R.: Efficient Detection of Distributed Constraint Violations. In: *Proc. of IEEE ICDE*. (2007) 1320–1324
3. Babcock, B., Olston, C.: Distributed Top-k Monitoring. In: *Proc. of ACM SIGMOD*. (2003) 28–39
4. Bertsekas, D.: *Nonlinear Programming*. Athena Scientific (1999)
5. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Gossip Algorithms: Design, Analysis, and Applications. In: *Proc. IEEE INFOCOM*. (2005) 1653–1664
6. Cormode, G., Garofalakis, M.: Sketching streams through the net: Distributed approximate query tracking. In *Proc. VLDB*. (2005) 13–24
7. Cormode, G., Garofalakis, M., Muthukrishnan, S., Rastogi, R.: Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In: *Proc. ACM SIGMOD*. (2005) 25–36
8. Cormode, G., Muthurikshnan, S., Yi, K.: Algorithms for Distributed Functional Monitoring. In: *Proc. ACM SODA*. (2008) 1076–1085
9. Das, A., Ganguly, S., Garofalakis, M., Rastogi, R.: Distributed set-expression cardinality estimation. In: *Proc. VLDB*. (2004) 312–323
10. Deshpande, A., Guestrin, C., Madden, S., Hellerstein, J. M., Hong, W.: Model-Driven Data Acquisition in Sensor Networks. In: *Proc. VLDB*. (2004) 588–599
11. Deshpande, A., Guestrin, C., Madden, S.: Using Probabilistic Models for Data Management in Acquisitional Environments. In: *Proc. CIDR*. (2005) 317–328
12. Dilman, M., Raz, D.: Efficient Reactive Monitoring. In: *Proc. IEEE INFOCOM*. (2001) 1012–1019
13. Jadbabaie, A., Lin, J., Stephen Morse, A.: Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control*. **48**(6) (2003) 988–1001

14. Kashyap, S., Ramamirtham, J., Rastogi, R., Shukla, P.: Efficient Constraint Monitoring Using Adaptive Thresholds. In: Proc. IEEE ICDE. (2006) 526–535
15. Kelly, F. P., Maulloo, A., Tan, D. K. H.: Rate Control for Communication Networks: Shadow Prices, Proportional Fairness, and Stability. Operational Research Society. **49**(3) (1998) 237-252
16. Keralapura, R., Cormode, G., Ramamirtham, J.: Communication-Efficient Distributed Monitoring of Thresholded Counts. In: Proc. ACM SIGMOD.(2006) 289–300
17. Kifer, D., Shai, B., Gehrke., J.: Detecting Change in Data Streams. In: Proc. VLDB. (2004) 180–191
18. Olston, C., Jiang, J., Widom, J.: Adaptive filters for continuous queries over distributed data streams. In: Proc. ACM SIGMOD. (2003) 563–574
19. Poosala, V., Ioannidis, Y.: Selectivity Estimation Without The Attribute Value Independence Assumption. In: Proc. VLDB. (1997) 486–495
20. Sharfman, I., Schuster, A., Keren, D.: A Geometric Approach to Monitoring Threshold Functions over Distributed Data Streams. In: Proc. ACM SIGMOD. (2006) 301–312